JOURNAL OF COMPLEXITY 8, 230–238 (1992)

$P_{\mathbb{R}} \neq NC_{\mathbb{R}}$

FELIPE CUCKER*

Dept. Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Barcelona 08028, Spain

Received November 14, 1991

In this note, we show the existence of sets of real numbers that can be decided in polynomial time for the Blum, Shub and Smale model of computation but cannot be decided in polylogarithmic parallel time using an arbitrary number of processors.

INTRODUCTION

In a recent paper (Blum *et al.*, 1989), a theory of computation over an arbitrary ring was devised that introduced ideas and methods from classical complexity and computability theories into the area of algebraic complexity (see von zur Gotham, 1988, for a survey of this latter subject). A special emphasis was placed on the case in which the ring is $\mathbb{R}$, the field of real numbers. The theory then reflects the kind of computations made in numerical analysis or computational geometry. For that special case, many basic results have been shown such as the existence of natural $NP_{\mathbb{R}}$-complete problems or universal machines.

In a subsequent work (Cucker and Torrecillas, 1991), the existence of natural $P_{\mathbb{R}}$-complete problems was also proved. But, just as the existence of $NP_{\mathbb{R}}$-complete problems left open the question of whether $P_{\mathbb{R}} = NP_{\mathbb{R}}$, the existence of $P_{\mathbb{R}}$-complete ones focuses interest on the question of whether polynomial time equals parallel polylogarithmic time (with a polynomial number of processors). In this paper we answer this question

* Partially supported by the ESPIRIT BRA Program of the EC under Contract 3075, Project ALCOM, DGICyT PB 89/0379, and UPC PR9014.

0885-064X/92 $5.00

negatively. Moreover, the method used does not depend on the number of processors used by the parallel model and readily extends to a very general result that allows us to separate several complexity classes over the reals. A last consequence of this result is the fact that $NP_{\mathbb{R}}$ is strictly included in $EXP_{\mathbb{R}}$.

1. Ground Tools and Notations

In the following we denote the direct sum $\oplus_{i}^{\infty}\mathbb{R}$ by $\mathbb{R}^{\infty}$. We recall that this direct sum is the set of sequences of real numbers having only a finite number of nonzero elements. Moreover, we define the *size* $|x|$ of an element $x \in \mathbb{R}^{\infty}$ as the largest i such that its i^{th} coordinate x_i is different from zero.

We recall from Blum *et al.* (1989) that a *real Turing machine* consists of an input space $\mathbb{R}^{\infty}$, an output space $\mathbb{R}^{\infty}$, and a state space $S = \mathbb{Z}^{+} \times \mathbb{Z}^{+} \times \mathbb{R}^{\infty}$, together with a connected directed graph whose nodes, labeled $1, \ldots, N$ (the set of different instructions), are of certain types and with associated functions. The internal content of S at time T is $(i, j, x_1, x_2, x_3, \ldots)$, where for $t = 1$ the input is in the x_s with s odd (thus we reserve the even coordinates to leave work space), and x_2 can denote the length of the input. The five types of nodes are as follows:

(1) *Exactly one input node:* node 1. Associated with this node is a next node $\beta(1)$.

(2) *Exactly one output node:* node N. Once it is reached the computation halts, the contents of the real part of S being considered as the output.

(3) *Computation nodes.* Associated with a node m of this type there are a next node $\beta(m)$ and a map g_m: $S \to S$. The g_m is of the form $g_m(i, j, x) = (i'(i), j'(j), x'(x))$, with $i'(i) = i + 1$ or $1, j'(i) = j + 1$ or 1, and x' is a polynomial or rational map.

(4) *Branch nodes.* There are two nodes associated with this node: $\beta^{+}(m)$ and $\beta^{-}(m)$. The next node is $\beta^{+}(m)$ if $x_1 \geq 0$ and $\beta^{-}(m)$ otherwise.

(5) *Move nodes or fifth nodes.* Each of these has a unique next node $\beta(m)$. If the current element of S is $(i, j, x_1, \ldots)$ it operates replacing x_j by x_i in the j^{th} place of the vector $\mathbb{R}^{\infty}$ in S.

An instantaneous description of any moment of the computation can be given by providing an element in S and the current node. The first one changes according to the function associated with the current node and the node itself according to the function β.

We also recall from Blum *et al.* (1989) that a machine M is said *to work in polynomial time* when there are constants $c, q \in \mathbb{Z}^{+}$ such that for every input $y \in \mathbb{R}^{\infty}$, M reaches its output node after at most $c(\text{size}(y))^{q}$ steps.

The class $P_{\mathbb{R}}$ is then defined as the set of all subsets of $\mathbb{R}^\infty$ that can be accepted by a machine working in polynomial time.

2. Parallel Models

There is no unified theory of parallel machines for the real numbers. However, research done in algebraic complexity extensively used circuits as models of computations either as a nonuniform model for getting lower bounds or combined with some uniformity condition when possible. A recent survey of that subject can be found in von zur Gathen (1986).

We introduce now a class of circuits together with a class of sets defined through them. They are equivalent to the arithmetical networks of von zur Gathen (1986).

Definition. An *algebraic circuit over the reals* with inputs in $\mathbb{R}^n$ is a finite directed graph $\mathcal{C}$, whose nodes have labels from $\mathbb{N} \times \mathbb{N} - \{0\}$, that satisfies the following conditions:

- There are exactly n nodes $v_{01}, \ldots, v_{0n}$ with first index 0, and they have no incoming edges
- all the other nodes v_{ij} are of one of the following types

(1) *arithmetic nodes:* they have an associated arithmetic operation $\{+, -, *, /\}$ and there exist l, k, r, m with $l, k < i$ such that their incoming edges are (v_{lr}, v_{ij}) and (v_{km}, v_{ij}).

(2) *constant nodes:* they have an associated real number γ and no incoming edges.

(3) *sign nodes:* they have a unique incoming edge (v_{km}, v_{ij}) with $k < i$.

To each node we inductively associate a function of the input variables in the usual way. We note that a sign node with input x returns 1 if $x > 0$ and 0 otherwise. Also, we call *depth* of the circuit the largest m such that we have nodes v_{mj}, and *size* of the circuit the total number of nodes.

Also, a circuit of depth d is *decisional* if there is only one node v_{d1} at level d, and it is a sign node. We finally define the *accepted set* of a decisional circuit to be the set $S \subset \mathbb{R}^n$ of the points whose image by the associated function is 1.

In order to get a uniform model of parallel computation we should endow families of circuits with some uniformity condition, but the most usually used in the Boolean case to define the class NC—the generation of the circuits by a machine working in logarithmic space—is meaningless now. One of the remarkable features of the theory of computation over the reals is given by the fact that to obtain complexity classes bounding the used space is irrelevant. Thus, in Michaux (1989) it is shown that any

recursive subset of $\mathbb{R}^\infty$ can be decided by a real Turing machine within linear space.

One possible way, then, of defining, uniformity is given by imposing the existence of a real Turing machine which, given input n, generates the n^{th} circuit in time which is polynomial in n. The complexity class $\text{PUNC}_{\mathbb{R}}^k$ defined by such families of circuits having polylogarithmic depth and polynomial size is an analog of the class PUNC defined in the boolean case, which contains NC (see Allender, 1988). Another possibility is to require both polynomial time and logarithmic space to the above mentioned real Turing machine. The later requirement allows to define a class more similar to NC.

One can also define models like PRAMs or PRTMs with a polynomial number of processors (real RAMs (see Preparata and Shamos, 1985) or real Turing machines) that work within polylogarithmic time and that communicate directly between them or via a shared memory, and obtain for every k their associated complexity classes $\text{PRAM}_{\mathbb{R}}^k$ and $\text{PRTM}_{\mathbb{R}}^k$. In the Boolean case, there is a large amount of work done showing the equivalence of those models and the cost of the simulations among them. For computations with real numbers, this is something waiting to be done.

Concerning the result we want to prove, there is no need, however, of using a particular model. The only feature we shall use is that, for all inputs of a given size n and at each moment of the computation, the actual configuration consists of a finite number of nonzero coordinates in the space state, and that at each computing step a new configuration is obtained from the present one modifying some of the coordinates of the space state (at most as many as the number of processors) replacing them by the result of operating (via one of $(+, -, *, /)$) on two other coordinates. These modifications may depend on a set of Boolean conditions (again at most as many as the number of processors) of the form $x \geq 0$, where x is the value of one of these coordinates. We observe that this is exactly what happens with the circuits above introduced (independently of any uniformity condition) or with any PRAM or PRTM. Therefore, our results will be valid for both the uniform and the nonuniform cases.

On the other hand, as we can expect, we have the following

THEOREM 2.1. *For every k, the classes* $\text{PUNC}_{\mathbb{R}}^k$, $\text{PRAM}_{\mathbb{R}}^k$, *and* $\text{PRTM}_{\mathbb{R}}^k$ *are contained in* $P_{\mathbb{R}}$.

3. THE THEOREM

We briefly recall some basic notions in algebraic geometry that will be useful to us in the sequel.

A set $V \subset \mathbb{C}^k$ is called an *algebraic set* when V is the set of all points in $\mathbb{C}^k$ satisfying a system of polynomial equations

$$\begin{aligned} f_1(X_1, \ldots, X_k) &= 0 \\ f_2(X_1, \ldots, X_k) &= 0 \\ &\vdots \\ f_r(X_1, \ldots, X_k) &= 0. \end{aligned}$$

Of course, all the polynomials belonging to the ideal generated by $f_1, \ldots, f_r$ also vanish on V. On the other hand, this ideal is called a *definition ideal* of V when all the polynomials vanishing on V belong to it. Hilbert's Nullstellensatz characterizes the ideals of $\mathbb{C}[X_1, \ldots, X_n]$ that are definition ideals of some algebraic set, which turn out to be the radical ones (see Fulton, 1969).

Also, an algebraic set V is said to be *reducible* when there exist two algebraic sets V_1 and V_2, both different from V, such that $V = V_1 \cup V_2$. It is a basic fact that a set is irreducible iff its definition ideal is prime.

In the sequel we are concerned with plane algebraic curves, i.e., curves in $\mathbb{C}^2$ given by a single polynomial in $\mathbb{C}[X, Y]$. More concretely, we deal with some Fermat curves which are given by polynomials of the form $X^d + Y^d - 1$, and we denote by $\mathcal{F}_d$ the set of its complex points and by $\mathcal{F}_d^{\mathbb{R}}$ its intersection with $\mathbb{R}^2$. We recall that such polynomials are irreducible (since they define algebraically nonsingular curves in the projective plane) and thus generate prime ideals in $\mathbb{C}[X, Y]$.

Let us now introduce the problem

$$\text{FER} = \{x \in \mathbb{R}^\infty \mid |x| = n \text{ then } (x_1, x_2) \in \mathcal{F}_{2^n}^{\mathbb{R}}\}$$

where, we recall $|x|$ stands for the size of x, and whose first property is given in the next result.

PROPOSITION 3.1. *The problem* FER *belong to* $\mathrm{P}_{\mathbb{R}}$.

Proof. The following algorithm

```
begin
  n := |x|;
  a := x_1;
  b := x_2;
  for i = 1 to n do
    a := a * a;
    b := b * b
  od
  if a + b = 1 then ACCEPT
    else REJECT
  fi
end
```

recognizes FER in linear time. ■

We can show our main result.

THEOREM 3.2. *For all $k \in \mathbb{N}$ and all function $f: \mathbb{N} \to \mathbb{N}$ there is no parallel machine accepting* FER *within time* $\log^k n$ *using* $f(n)$ *processors.*

Proof. Let us assume that there is a parallel machine M, as in the statement that solves FER. For any n and any input $(x_1, \ldots, x_n)$ of size n consider the tree of all possible configurations of the machine. For the sake of simplicity, we suppose that $x_3 = \cdots = x_n = 1$ without loss of generality. Each configuration can be described by a point in the state space $\mathbb{R}^N$, where now N is a fixed bound that only depends on n.

At each step of the computation we modify some of the coordinates, replacing them by the resulting of operating (via one of $(+, -, *, /)$) on two other coordinates. Those modifications can depend on Boolean conditions of the form

$$Q_i(x_i, x_2) \geq 0$$

—where $Q_i(x_1, x_2)$ is the content of cell i and is a rational function in x_1 and x_2. Those Boolean conditions produce a branching in our tree of configurations. Moreover, since the number of processors is bounded by $f(n)$, the fan-out of each node in the tree of configurations is bounded by $2^{f(n)}$. After $\log^k n$ steps, we could have a large (but finite) number of leaves that are accepting or rejecting leaves, and FER is the union of the sets of inputs for which the computation leads to an accepting leaf.

For each one of those accepting leaves, the final configuration will consist of at most N rational functions in x_1 and x_2 whose numerator and denominator have a degree which is bounded by $2^{\log^k n}$, since the depth of the tree is $\log^k n$. Thus, all the rational functions $Q_i(x_1, x_2)$ appearing in the Boolean conditions above mentioned have the same bounds for the degrees. We conclude that the set of inputs that are led to a given leaf can be characterized by a finite system of inequalities of the form

$$\bigwedge_{i=1}^{s} Q_i(X_1, X_2) \leq 0 \wedge \bigwedge_{i=s+1}^{t} Q_i(X_1, X_2) < 0,$$

where t is bounded by $f(n) \log^k n$. By clearing denominators we can replace the rational functions by polynomials with the same (actually twice the) bound for the degrees that we had for the rational functions. Also, expressing an inequality like

$$F(X_1, X_2) \geq 0$$

as the disjunction

$$F(X_1, X_2) = 0 \vee F(X_1, X_2) > 0$$

and then distributing, we can describe FER as a union of sets given by systems of polynomial inequalities of the form

$$\bigwedge_{i=1}^{s} F_i(X_1, X_2) = 0 \wedge \bigwedge_{j=1}^{t} G_j(X_1, X_2) > 0.$$

Now, since the curve $\mathcal{F}_{2^n}^{\mathbb{R}}$ is infinite, one of those sets must contain an infinite number of points of the curve. Since the set described by the G_j's is open, it must be nonempty, and then it defines an open subset of $\mathbb{R}^2$. But $\mathcal{F}_{2^n}^{\mathbb{R}}$ is a curve, and therefore we must have $s > 0$.

Finally, all the polynomials F_i, $i = 1, \ldots, s$, vanish on that infinite subset of the curve and, thus, in a 1-dimensional component of the curve. But, since the curve is an irreducible one, this implies that every F_i must vanish on the whole curve. Using the fact that the ideal $(X_1^{2^n} + X_2^{2^n} - 1)$ is prime (and, a fortiori, radical), we conclude that all the F_i are multiples of $X_1^{2^n} + X_2^{2^n} - 1$ which is impossible since their degree is bounded by $2^{\log^k n}$, which is strictly smaller than 2^n. ■

The main argument in the preceding theorem can be used to get a more general result. Before stating it, let us recall that a total function $f: \mathbb{N} \to \mathbb{N}$ is said to be *time constructible* when there is a real Turing machine that on input n computes $f(n)$ within time $O(f(n))$. We then have

THEOREM 3.3. *Let $t, t' : \mathbb{N} \to \mathbb{N}$ be two time bounds, t' time constructible, and assume that $t' \in \omega(t)$. Then there is a set $S \subset \mathbb{R}^\infty$ which is recognized by a real Turing machine in time t' but cannot be recognized in parallel time t with any number of processors.*

Proof. Just use the argument of Theorem 3.2 with the set

$$\{x \in \mathbb{R}^\infty \mid |x| = n \text{ then } (x_1, x_2) \in \mathcal{F}_{2^{t'(n)}}^{\mathbb{R}}\}.$$ ■

COROLLARY 3.4. *(i) The complexity classes defined by parallel time $\log^k n$ are different for different k.*

(ii) The complexity classes defined by parallel time n^k are different for different k.

Part (ii) of the corollary above has an immediate consequence concerning the classes $\mathrm{NP}_{\mathbb{R}}$ and $\mathrm{EXP}_{\mathbb{R}}$, which—we recall—is defined as the union for $k \geq 1$ of the subsets of $\mathbb{R}^\infty$ accepted by real Turing machines in time $O(c^{n^k})$ for some $c > 1$.

PROPOSITION 3.5. *The inclusion $\mathrm{NP}_{\mathbb{R}} \subset \mathrm{EXP}_{\mathbb{R}}$ is strict.*

Proof. We recall from Heintz *et al.* (1990) or Renegar (1989) that the $NP_{\mathbb{R}}$-complete problem 4FEAS is solvable in parallel polynomial time with an exponential number of processors, and thus, all $NP_{\mathbb{R}}$ problems can be solved within the same resources. But by corollary 3.4(ii) we now that there is a problem in $EXP_{\mathbb{R}}$ that does not have this property and this concludes the proof. ■

REMARKS 3.6. (i) FER is not a natural problem, but just a technical way of getting the desired separations of classes. However, as a consequence of Theorem 3.2, we can now assert that the $P_{\mathbb{R}}$-complete problems exhibited in Cucker and Torrecillas (1991) have no $NC_{\mathbb{R}}$ algorithms.

(ii) As one can expect, the preceding argument cannot be applied as it stands in the discrete case. The main obstruction is the lack of infinite points of a given size in the algebraic curves.

(iii) On the other hand, the same proof applies for machines over $\mathbb{C}$ (see Smale, 1990, or Shub, to appear, for such machines).

ACKNOWLEDGMENT

The results in this paper arose from a talk with Mike Shub at Berkeley and subsequently an e-mail dialog. The author is greatly indebted to him.

References

ALLENDER, E. (1986), Characterizations of PUNC and precomputation, *in* "13th ICALP," Lecture Notes in Computer Science, Vol. 226, pp. 1–10, Springer, Berlin/New York.

BLUM, L., SHUB, M., AND SMALE, S. (1989), "On a theory of computation and complexity over the real numbers: *NP*-completeness, recursive functions and universal machines, *Bull. Amer. Math. Soc.* **21**(1), 1–46.

CUCKER, F., AND TORRECILLAS, A. (1991), Two P-complete problems in the theory of the reals *in* "18th ICALP," Lecture Notes in Computer Science, Vol. 510, pp. 556–565, Springer, Berlin/New York.

FULTON, W. (1969), "Algebraic Curves," Benjamin, New York.

VON ZUR GATHEN, J. (1986), Parallel arithmetic computations: A survey, *in* "Proceedings, 12th Int. Symp. Math. Found. Comp Sci., Lecture Notes in Computer Science, Vol. 233, pp. 93–112, Springer-Verlag, Berlin/New York.

VON ZUR GATHEN, J. (1988), Algebraic complexity theory, *Annual Rev. Comput. Sci.* **3,** 317–347.

HEINTZ, J., ROY, M.-F., AND SOLERNO, P. (1990), "Sur la complexité du principe de Tarski–Seidenberg, *Bull. Soc. Math. France* **118,** 101–126.

MICHAUX, C. (1989) Une remarque à propos des machines sur $\mathbb{R}$ introduites par Blum, Shub et Smale, *C.R. Acad. Sci. Paris Sér-I Math.* **309,** 435–437.

PREPARATA, F. P., AND SHAMOS, M. I. (1985), "Computational Geometry: An Introduction," Texts and Monographs in Computer Science, Springer-Verlag, Berlin/New York.

RENEGAR, J. (1989), On the computational complexity and geometry of the first order theory of the reals, Parts I, II, and III. Technical Reports 853, 854, and 856, Cornell University.

SMALE, S. (1990), Some remarks on the foundations of numerical analysis, *SIAM Rev.* **32**(2), 211–220.

SHUB, M. (to appear), On the work of Steve Smale on the theory of computation, *in* "Proceedings of the Smalefest" (M. Hirsch, J. Marsden, and M. Shub, Eds.).